"Express Mail" mailing label number:
EV401038533US

PROCESSING OF PROCESSOR PERFORMANCE STATE INFORMATION

Richard Alan Hamersley

CROSS-REFERENCE TO RELATED APPLICATION(S)

BACKGROUND

Field of the Invention

[1001] This invention relates to computer systems and more particularly to aspects of power management and BIOS.

Description of the Related Art

[1002] Computing systems are information handling systems which are designed to give independent computing power to one or more users. Computing systems can be found in many forms including, for example, mainframes, minicomputers, workstations, servers, personal computers, internet terminals, notebooks and embedded systems. Personal computer (PC) systems include desk top, floor standing, or portable versions. A typical PC system is a microcomputer that includes a microprocessor, associated memory and control logic (typically on a system board) and a number of peripheral devices that provide input and/or output (I/O) for the system.

[1003] Computing systems typically include a set of built-in software routines called the basic input/output system (BIOS). The BIOS is a software interface between the system hardware and the operating system software. The BIOS facilitates programmer and user interaction with the system hardware. The BIOS is often referred to as firmware. Like software, the BIOS is a set of instructions to the computer's microprocessor. The BIOS is commonly coded using, for example, assembly language, and stored onto a non-volatile memory such as a ROM (Read Only Memory) or a PROM (Programmable ROM) such as an EPROM (Erasable

PROM), an EEPROM (Electrically Erasable PROM), a flash RAM (Random Access Memory) or any other type of memory appropriate for storing BIOS.

[1004] The BIOS controls several important functions of personal computer systems. For instance, the BIOS performs various functions at power up, including testing and initializing memory, inventorying and initializing the system, and testing the system. These functions at power up are referred to as "system boot" or "booting the system" and can occur every time the system powers up or is reset. One part of power up is referred to as Power On Self Test (POST). The POST routines include testing various aspects of the system.

[1005] The BIOS also controls keystroke interpretation, display of characters, and communication via the PC ports. The operating system and application programs of a computer system access the BIOS rather than directly manipulating I/O ports, registers and control words of the specific system hardware. BIOS can be accessed through an interface of software interrupts and contains a plurality of entry points respectively corresponding to the different interrupts.

[1006] Another function supported by BIOS relates to processor performance states. Microprocessors utilized in mobile applications, i.e., those used in battery powered systems, are particularly sensitive to power considerations. That is in part due to the small, densely packed system construction that limits the ability of the mobile computer system to safely dissipate the heat generated. A particular processor may operate at a number of different performance levels to achieve power management goals.

[1007] Appropriately monitoring and controlling the processor's operating parameters is important to optimizing the computer's performance, power consumption, and in the case of a portable or notebook computer, battery life. One of the current power management techniques utilized provides a computer system that has multiple performance states. The computer system determines an appropriate performance state based on system utilization or other information, such as temperature or battery life, and adjusts the performance of the system accordingly. Each performance state is made up of a particular voltage and operating frequency. The performance states may be stored in performance state tables, which specify a

plurality of voltage and frequency combinations for a particular processor. In one approach, e.g., the PowerNow!™ technology from Advanced Micro Devices, a driver utilizes the performance state tables to dynamically change performance states and saves power by operating only as fast as required by the current system utilization.

[1008] The BIOS programmer has to manually build each performance state table from information provided, e.g., in processor data sheets. Because BIOS typically has to support several different processors over the life of a product, many performance state tables may be included in a BIOS version. However, certain performance states supported by one processor may not be supported by another processor. In fact, processors may be nominally the same but be built by slightly different processes resulting in different voltages and frequencies being supported. It is undesirable to revise BIOS frequently to support additional performance states. On the other hand it is may be unwieldy for BIOS to support large numbers of performance states.

[1009] Accordingly, it would be desirable to provide a way for BIOS to effectively handle a large number of performance states without affecting runtime resources. In addition, it would be desirable to be able to suitably support a processor when none of the performance state tables in BIOS match the processor in the system.

SUMMARY

[1010] Storing multiple performance state tables for various processors in a disposable memory segment and storing one of the multiple performance state tables in a runtime memory segment, allows the performance state table footprint in runtime memory to be reduced while facilitating a BIOS that supports multiple processors. The performance state table stored into the runtime memory segment satisfies to some degree one or more criteria that correspond to the processor of the system in which the BIOS resides. However, some embodiments of the invention provide for a default performance state table or provide for dynamic generation of a performance state table, typically when a performance state table in the disposable memory segment does not satisfy the processor criteria.

[1011] According to some embodiments of the invention, a method of operating a computer system comprises determining if any performance state data stored in the

computer system in a first area of memory specifying performance states for a plurality of processors is associated with a processor being utilized in the computer system. The method also comprises copying the portion of the performance state data into a second area of memory if a portion of the performance state data is found to be associated with the processor being utilized in the computer system.

[1012] According to some embodiments of the invention, a computer program product stored on computer readable medium is operable in a computer system to determine whether a match exists between a processor being utilized in the computer system and performance state information for a plurality of processors. In addition, the computer program product is operable to generate performance state data after no match is found to exist.

[1013] According to some embodiments of the invention, an apparatus comprises means for determining if any performance state data specifying performance states for a plurality of processors, which is stored a first area of memory of a computer system utilized for system boot, is associated with a processor being utilized in the computer system. The apparatus also comprises means for copying a portion of the performance state data into a second area of memory if the portion of the performance state data is found to be associated with the processor being utilized in the computer system.

[1014] According to some embodiments of the invention, an apparatus comprises means for evaluating information in a plurality of performance state tables, each of the performance state tables having information relating to a performance state of a processor. The apparatus also comprises means for determining if a near match exists and utilizing one of the performance state tables that is a near match to the processor if the near match exists, if a match is not found between one of the performance state tables and a processor being utilized in the computer system.

[1015] According to some embodiments of the invention, an apparatus comprises means for determining if any performance state data stored in the computer system specifying performance states for a plurality of processors is associated with a processor being utilized in the computer system. The apparatus also comprises means

for generating performance state data if none of the performance state data is associated with the processor being utilized in the computer system.

[1016] According to some embodiments of the invention, a computer program product stored on computer readable medium operable in a computer system to copy one of a plurality of performance state data associated with respective processors from a disposable memory segment into a runtime memory segment. The computer program product is also operable to generate an advanced configuration and power interface object based at least in part on the copied performance state data.

[1017] According to some embodiments of the invention, a computer system comprises memory that hosts a basic input/output system (BIOS) that selects a first of a plurality of performance state information from a first area of memory into a second area of memory. The first area of memory is to be utilized during power on self test (POST) processing by the BIOS, and the second area of memory is to be used for a runtime memory segment of BIOS. The computer system also comprises a processor coupled with the memory.

[1018] These and other aspects of the described invention will be better described with reference to the Detailed Description and accompanying Figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[1019] The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings.

[1020] Figure 1A illustrates performance state tables hosted in runtime memory segment according to realizations of the invention. Figure 1B illustrates a runtime memory space segment hosting a PSB with a selected PST according to realizations of the invention.

[1021] Figure 2 illustrates an exemplary performance state block according to realizations of the invention.

[1022] Figure 3 illustrates an exemplary performance state table according to some realizations of the invention.

[1023] Figure 4 illustrates an exemplary CPUID field according to realizations of the invention.

[1024] Figure 5 illustrates an exemplary flowchart for storing a PST according to some realizations of the invention.

[1025] Figure 6 illustrates a conceptual diagram of a BIOS dynamically generating ACPI objects according to some realizations of the invention.

[1026] Figure 7 depicts an exemplary computer system according to some realizations of the invention.

[1027] The use of the same reference symbols in different drawings indicates similar or identical items.

DETAILED DESCRIPTION

[1028] The description that follows includes exemplary systems, methods, techniques, and data structures that embody techniques of the present invention. However, it is understood that the described invention may be practiced without these specific details. In other instances, well-known protocols, structures and techniques have not been shown in detail in order not to obscure the invention.

[1029] Figure 1A illustrates performance state tables hosted in runtime memory segment according to realizations of the invention. In Figure 1A, a memory space 101 includes a runtime memory space segment 103. The runtime memory space segment 103 hosts multiple performance state tables (PSTs). For example, the runtime memory space segment 103 is 64 kilobytes of memory space, identified as a segment 0xF000. A BIOS can load multiple PSTs into the runtime memory space segment, thus allowing a single BIOS to support various processors that correspond to the PSTs. The BIOS selects the appropriate one of the loaded PSTs for the host processor and modifies the other PSTs in order to avoid using them (e.g., scrambling the headers of the PSTs).

[1030] Figure 3 illustrates an exemplary performance state table according to some realizations of the invention. A PST 300 includes a header 301, which includes a CPUID field, MaxFID field, StartVID field, and a front side bus (FSB) clock speed field. The FSB clock speed field refers to the speed of a clock utilized by an external bus coupled to the processor. The processor may use that clock to generate internal clock signals. The FID (frequency identifier) fields represent the processor clock frequency multipliers The V fields indicate voltages that correspond to the processor. The speed of the processor clock is determined by multiplying the FSB clock by the multiplier represented by the current FID value. The MaxFID field indicates the highest clock multiplier at which the processor can operate, and the StartVID field indicates a coded value that corresponds to the processor voltage. The CPUID field identifies a processor that corresponds to the associated PST. In addition, the PST 300 includes a plurality of voltage and frequency combinations 303 that together specify a performance state for the processor. The frequencies may be represented as FIDs in the PST as illustrated in the drawing.

[1031] Figure 4 illustrates an exemplary CPUID field according to realizations of the invention. A CPUID field 400 includes a family field 401, a model field 403, and a step field 405. The family field 401 indicates a processor family (e.g., AMD-K6®, AMD Athlon, AMD Athlon 64, AMD Opteron, etc.). The model field 403 indicates a processor model (e.g., FX, XP, MP, etc.). The step field 405 indicates stepping bits, which represents a step or revision of a die. The CPUID field illustrated in Figure 4 is illustrative and not meant to be limiting upon the invention. Various realizations of the invention include fewer fields or less fields than illustrated in Figure 4 (e.g., an extended family field).

[1032] If multiple PSTs are placed in the runtime memory segment, a single BIOS can support multiple processors. Supporting multiple processors with a single BIOS reduces maintenance burdens and increases efficiency of product maintenance and production. However, consumption of valuable memory space in the runtime memory segment, which is limited in size, can be reduced while still supporting different processors. In various realizations of the invention, a BIOS performs varying levels of screening of PSTs for multiple processors. Varying levels of screening reduces the amount of the runtime memory segment consumed by PSTs.

[1033] In some embodiments of the invention, a BIOS loads multiple PSTs into a "throw away" or disposable memory segment. The BIOS screens the throw away memory segment for an appropriate PST. The BIOS constructs a simple single-PST performance state block (PSB) at the appropriate address boundary in a runtime memory space segment. Figure 2 illustrates an exemplary performance state block according to realizations of the invention. A performance state block 200 includes multiple performance state tables (PSTs) as well as a PSB header. The PSB header indicates information that aids in navigating the PSTs. In this example, the PSB header is a 16 byte header. The exemplary 16 byte header includes a PSB signature (e.g., character string that identifies the BIOS), a table version, flags (e.g., indicating mobile, desktop, etc.), settling time, reserved, and an indication of the total number of PSTs in the PSB. Various realizations include implement a PSB differently (e.g., information in the exemplary PSB header is separate from a grouping of PSTs, more or less information is included in the PSB header, the PSB header may be a different size, etc.).

[1034] The appropriate PST will match a set of parameters corresponding to the host processor. For example, those parameters include the CPUID, MaxFID, StartVID, and FSB clock speed. Other embodiments may utilize other parameters in place of, or in addition to those named.

[1035] If a match is found with a PST in the throw away segment, the BIOS copies that PST's information to the single PST runtime PSB. Screening available PSTs that are stored in a throw away memory segment for a parameter matching PST reduces the footprint for supporting a PSB in the runtime memory significantly. To illustrate the reduction, in some embodiments of the invention, the memory footprint can be reduced from over 500 bytes to a fixed size of only 56 bytes. The 56 byte footprint supports up to 16 power states (at 2 bytes per power state). By placing the available PSTs in the throw away memory segment, a BIOS can support a large number of pre-defined PSTs without consuming valuable memory space in runtime memory space segment.

[1036] In some embodiments, the PST support code and tables are located in the throw away memory segment of the BIOS that is only available during power-on self-

test (POST). The POST is a period of time during which BIOS initializes and tests the hardware resources of a personal computer.

[1037] Figure 1B illustrates a runtime memory space segment hosting a PSB with a selected PST according to realizations of the invention. A memory space 105 includes a disposable memory segment 107 and a runtime memory segment 109. The disposable memory segment 107 may include multiple PSBs. A BIOS 111 screens the PSBs in the disposable memory segment 107 and selects a PST from the PSBs that matches a given set of processor parameters. The BIOS 111 stores the selected PST into a PSB constructed in the runtime memory segment 109. The BIOS may construct the runtime PSB before selecting a matching PST, after selecting a matching PST, etc. In some embodiments, the PSTs are split into two basic groups – PSTs for mobile processors and PSTs for low-power or small form factor (SFF) desktop processors. The small form factor processors are low-power desktop processors. For example, one PSB in disposable memory includes PSTs for mobile processors and another PSB includes PSTs for SFF processors. The mobile PSTs may also be split into various groups corresponding, e.g., to low-voltage (e.g. < 1.2V core voltage) processors and regular mobile processors.

[1038] Separating PSTs into PSBs facilitates partitioning groups of PSTs according to some design criteria. In scanning for a parameter matching PST, in some embodiments, the BIOS determines which PSB is appropriate for the platform and scans through the multiple PSTs in that PSB looking for a PST that exactly matches the processor parameters. For example, a low voltage mobile PSB may be provided that includes PSTs that are utilized by low voltage mobile processors. When BIOS searches for an appropriate PST, that low voltage mobile processor PSB is the only PSB searched if the platform in which BIOS resides supports low voltage processors. Various realizations of the invention base selection of a PSB on various criteria (e.g., revision of a northbridge chip in a chipset).

[1039] In some embodiments, there is only one PSB in the disposable memory segment that contains all of the pre-defined PSTs. Alternatively, the BIOS may search all PSBs. After a match is found, the BIOS constructs a PSB block in the space-limited runtime area of the BIOS that contains only the PST that matches the

processor being utilized in the system. As stated above, this reduces the footprint required for PSTs in the runtime area of BIOS since only a single PST is placed in the runtime memory segment. Note that other appropriate data structures may also be used to store performance state information instead of PSBs or PSTs. Furthermore, a BIOS may select multiple PSTs to load into the runtime PSB.

[1040] If, however, an exact match is not found, the BIOS can determine if a "near match" exists. Figure 5 illustrates an exemplary flowchart for storing a PST according to some realizations of the invention. A search of the PSTs for an exact match, as described above, is performed in 501. If an exact match is found, the PST that is the exact match is copied over into the runtime memory segment in 503. If an exact match is not found, the PSTs are again searched in 505, except for this pass, the compare algorithm determines if a near match exists between the processor (or processors) being utilized in the system and the PST header. In some embodiments of the invention, a near match is defined as a match in which the stepping bits in the CPUID are ignored. Thus, even if an exact match cannot be found, a near match can be utilized. If a near match is found, then the near match PST is copied over to the runtime memory segment in 507. Use of a near match simplifies BIOS maintenance in that it is not critical that BIOS be updated for each processor change if the change simply involves the stepping bits.

[1041] Even if no near match exists, in some circumstances, BIOS can automatically generate the PST. Note that for certain processors, e.g., small form factor processors utilized in low power desktop implementations, the power states utilize a constant voltage and vary only the frequency. This means that the PST for such a processor includes only variable frequency information. The PST can be automatically generated because the PST is constructed in runtime memory. That is especially true of PSTs that only include variable frequency. Generating PSTs automatically when an exact match is not found allows support for processors even if a matching PST cannot be found.

[1042] Referring to Figure 5, an exemplary flow diagram illustrates automatic generation of a PST. If at 502 an exact match was not found, and if at 506 a near match was not found, the BIOS can automatically generate a frequency only PST in

509 in the runtime memory segment. In some embodiments, a PST is automatically generated by determining the upper and lower bounds of the frequency and then generating an appropriate number of intermediate frequencies spaced at appropriate intervals.

[1043] For example, the lower bound for a particular system may be 5 times the FSB clock speed. The lower bound may be determined according to the type of system or processor, or may be information available in the processor or system directly. The upper bound is indicated by the MaxFID field or from some other information provided in the processor or system. The intermediate frequencies may be spaced equally between the upper and lower bound, or according to some other metric. Once the upper bound, lower bound, and intermediate frequencies have been determined, the BIOS can build the PST in the runtime memory segment. Thus, processors can be supported even if the BIOS has not been updated to specifically support that processor. Although Figure 5 illustrates automatic generation of PSTs subsequent to checking for an exact match and a near match, in some embodiments, automatic generation of PSTs may follow a different flow (e.g., after checking for an exact match and recognizing an SFF platform).

[1044] While changing performance states may be supported in drivers in certain embodiments, in other embodiments, the operating system (e.g. Windows XP™) requires Advanced Configuration and Power Interface (ACPI) objects to support changing between various performance states. One approach to such a scenario is to provide separately maintained PSTs and ACPI objects. However, another approach is for these objects to be dynamically generated during POST rather than appearing statically in memory, which can potentially make special methods unnecessary, since static objects require a complex set of methods to figure out which table in the ACPI space to use. The memory footprint for the static implementation is also much larger, however the footprint size is not the major consideration that it is with PST tables in runtime BIOS memory since ACPI memory space is typically at or near the top of installed system memory, above the 1 MB address. Accesses to ACPI memory space are not constrained to the 64-Kbyte segment size of the runtime PSB. The ACPI memory space is practically limitless, so the BIOS developer is not as limited as with the memory space for the runtime PSB.

[1045] Figure 6 illustrates a conceptual diagram of a BIOS dynamically generating ACPI objects according to some realizations of the invention. The BIOS first builds in 601 a runtime PSB data structure 602 having a single PST as described above. The build in 601 uses as inputs a small form factor PSB 603, a mobile PSB 605, a low voltage mobile PSB 607, and/or a default PST template 609 as appropriate for the platform. The BIOS build in 601 is performed during POST and data structures 603-609 are static structures in the throw away memory segment. The default.PST template 609 is a table of processor VID codes sorted by clock multiplier value from lowest to highest. The default PST template 609 includes two columns: a first column includes whole number clock multipliers (e.g., 3x, 4x, 5x, 6x, etc.); and a second column includes fractional clock multipliers (e.g., 5.5x, 6.5x, 10.5x, etc.). Various realizations of the invention utilize values in either or both columns. The BIOS defines a _PSS object containing a number of (e.g. five) power states (P0, ..., P4). The ACPI 2.0 specification defines the _PSS object as an optional object that indicates a variable number of supported processor performance states. Additional details about the object are provided in section 8.3.3.2 of the ACPI specification. In some embodiments of the invention, the _PSS object is a fixed size object and the number of performance states is determined by the BIOS programmer at BIOS build time. The number of pre-defined performance states is independent of the actual number of performance states in the PST that supplies the performance state information. Those five power states are mapped to five of the power states in the runtime PST. In some embodiments of the invention, the BIOS dynamically constructs the _PSS object, hence the number of performance states in the dynamically constructed _PSS object can be the same as the number of performance states in the corresponding PST. The data for each _PSS state is then generated from information in the PST header and from the mapped power state in the runtime PST in 611. While in some embodiments an intermediate _PSS object is generated (e.g., for programming convenience), in other embodiments generation of the intermediate _PSS object is skipped. Finally the BIOS pokes valid _PSS data in 615 into the actual ACPI_PSS object to generate the ACPI_OBJECT 617. Various realizations of the invention will generate a different ACPI object in accordance with the ACPI specification (e.g., a differentiated system description table (DSDT) may include the _PSS object; a secondary system description table (SSDT) may include the _PSS

object, etc.). Generating ACPI objects, as described, facilitates efficient updating of ACPI objects by adding new PSTs.

[1046] Note that mapping of the runtime PST data into a predefined number of power states requires selectively including a subset of the available power states. Such selective inclusion can be implemented in a number of ways. Generally, the goal is to determine the maximum and minimum power state and fill the middle with power states spaced approximately evenly between the maximum and minimum.

[1047] The described invention may be provided as a computer program product, or software, that may include a machine-readable medium having stored thereon instructions, which may be used to program a computer system (or other electronic devices) to perform a process according to the present invention. A machine readable medium includes any mechanism for storing or transmitting information in a form (e.g., software, processing application) readable by a machine (e.g., a computer). The machine-readable medium may include, but is not limited to, magnetic storage medium (e.g., floppy diskette); optical storage medium (e.g., CD-ROM); magneto-optical storage medium; read only memory (ROM); random access memory (RAM); erasable programmable memory (e.g., EPROM and EEPROM); flash memory; electrical, optical, acoustical or other form of propagated signal (e.g., carrier waves, infrared signals, digital signals, etc.); or other types of medium suitable for storing electronic instructions.

[1048] Figure 7 depicts an exemplary computer system according to some realizations of the invention. A computer system 700 includes a processor unit 701 (possibly including multiple processors). The computer system 700 also includes a system memory 707A – 707F (e.g., one or more of cache, SRAM DRAM, RDRAM, EDO RAM, DDR RAM, EEPROM, etc.), a system bus 703 (e.g., LDT, PCI, ISA, etc.), a network interface 705 (e.g., an ATM interface, an Ethernet interface, a Frame Relay interface, etc.), and a storage device(s) 709A – 709D (e.g., optical storage, magnetic storage, etc.). Realizations of the invention may include fewer or additional components not illustrated in Figure 7 (e.g., video cards, audio cards, additional network interfaces, peripheral devices, etc.). The processor unit 701, the storage device(s) 709A – 709D, the network interface 705, and the system memory 707A –

707F are coupled to the system bus 703. The system memory 707A – 707F depicted in Figure 7 embodies a BIOS in accordance with the preceding description. For example, the embodied BIOS selects a PST from disposable memory and stores it in a runtime PSB. Although, the system memory 707A – 707D is depicted as coupled with the system bus, one or more of the system memory 707A – 707F may have a more direct connection with the processor 701 (e.g., DMA channel, on-chip, etc.).

[1049] While circuits and physical structures are generally presumed, it is well recognized that in modern semiconductor and design fabrication, physical structures and circuits may be embodied in computer readable descriptive form suitable for use in subsequent design, test, or fabrication stages as well as in resultant fabricated semiconductor integrated circuits. Accordingly, claims directed to traditional circuits or structure may, consistent with particular language thereof, read upon computer readable encodings and representations of same, whether embodied in media or combined with suitable reader facilities to allow fabrication, test, or design refinement of the corresponding circuits and/or structures.

[1050] The description of the invention set forth herein is illustrative, and is not intended to limit the scope of the invention as set forth in the following claims. Other variations and modifications of the embodiments disclosed herein, may be made based on the description set forth herein, without departing from the scope of the invention as set forth in the following claims.